

AUSTRALIAN OS9 NEWSLETTER
Newsletter of the National OS9 User Group



EDITOR : Gordon Bentzen

HELPERS : Bob Devries and Don Berrie

SUPPORT : Brisbane OS9 Level 2 User Group.

The new graphics which appears above is the handywork of one of our members, Peter Hughes. Peter designed this to be used as a letterhead for user group correspondence, and I think it looks great. Peter suggested that we include this sample and invite comment or other submissions or ideas. We also have a coloured version produced on Peter's colour printer, and he plans to photograph the CoCoMax screen so that we can have this logo printed onto T shirts. I believe there are a number of places that will enlarge a colour photo and print it on your T shirt for around \$12

Perhaps you could let us know if you would be interested in such a T-shirt, not that we want to go into business in this area, but we should be able to arrange a purchase deal somewhere which would enable us to supply printed T-shirts at a reasonable price. Does anybody have a contact in this area? What do you think?

The membership of the National OS-9 group continues to increase. Currently, membership stands at fifty three (53) with a couple of other recent enquiries. In this newsletter we provide information, program listings and reviews of hardware and software. We will also include advertisements free to our members, no we will not accept paid adds from commercial enterprises or businesses, but may include information about products available where we feel this would be of service to our members.

We will also include a Trading Post section if you have computer hardware or software for sale, or if you would like us to list something in the "Wanted" category. Any software listed for sale MUST of course be an Original issue for which, you have a legal right to sell.

The CoCo4 :- We have a little more information on the rumoured CoCo4 which comes to us via a system variously called News or Netnews. This is sort of like a global bulletin board. The letters are broadcast across the network for inspection by anyone who wants to do so. Two of the relevant newsgroups are "comp.os.os9" and "comp.sys.6809".

It appears that there are TWO independent projects underway to produce a cheap 68k/6809 computer using OS9, and retaining compatibility with CoCo OS9 (but not RSDOS). I have been advised very recently that one of these projects are being undertaken by Frank Hogg Labs. Another item gleaned from the list is that the OS9 super-guru Kevin Darling, with others, is producing an upgrade to CoCo Level 2 OS9, and this should be ready in the next few months. This effort is independent of Tandy and Microware, both of whom have given up on the CoCo.

At least one of the CoCo4 projects is being referred to as the KMA/CC4, not sure of the origin of the "KMA" but someone on the NET suggested it was "Kill Most Amigas/Ataris/Apples".

The following is a posting to the Net by Kevin Darling, dated Jan 17th 1990,
"Friends, I've been reading the list, but too busy to answer each time. So I thought I'd throw everything in at once <grin>.

KMA/CC4 - There are still two machines being worked on. details/diffs unfortunately cannot be released yet because each company is in competition with the other; and at least one is clamping up. In fairness I'll therefore await their permission. (BTW, that goes for L-II upgrade speculations... just be patient and be wary of "facts.")

However, some features on each/both machines should be mentioned in order to answer some comments/speculations made on the nets.

AUSTRALIAN OS9 NEWSLETTER

Any 6809 card would have a header for plugging in Disto's one-meg upgrade cpu board. There might indeed be sockets for the extra RAM, but do not forget that the main \$\$ part of that upgrade is the cpu board which has the external DAT to extend the GIME's MMU. Sockets alone are not enough!! Sure, a 1-meg DAT could be built onto the 6809 card, but they see no reason to zap Disto's market, nor to force people to throw away a one-meg upgrade kit they may already have bought, NOR to make someone pay extra for something they might not want/need.

Re: comparisons to the Amiga. I own one. It would make a nice OS9 machine, but its not easily expandable (except the A2000), and doesn't have a standard floppy controller. Under AmigaDOS, the minimum useable system is 1-meg and two drives anyway, which is about \$1000 for an A500. Go to a 68020/030 and we're talking A2000 and a total \$4500! That's affordable?

Compare that to one of the KMAs selling for under \$600, with 320-256 color mode (\$300 extra on the Amiga), 640-16 color mode, DMA stereo output AND inputs (extra \$\$ on Amiga), using IBM serial mouse and/or your old CoCo mouse, CoCo joysticks, hidens IBM-style (read: normal) disk controller, MIDI port, dual serial/parallel ports, and OS9 in ROM. Plus optional 68020/030 cards for a couple of thou\$ and less than the equivalent optional Amiga card. And a bus-based architecture for limitless expansion.

Sure, the Amiga has a glitter right now. That helps at times. On the other hand, their programs "cheat" and directly manipulate screen memory. Which means that they cannot move to machines where the video is handled TOTALLY away from the main cpu, by ANOTHER cpu dedicated to such a task... such as you'll be able to do with the KMA. Nor are they able to upgrade to future video cards that might have a 43020 or other hot gfx chip. Etc etc. Their hardware is falling way behind now however, and they know it.

OS9ers are willing to take the long view. Many programs written almost 10 years ago, and they know that most will still work 10 years from now, even as the hardware art advances (which it most assuredly does and will). If you're into instant and temporary gratification, get a Nintendo for your games, and a PCclone for biz programs. That's perfectly fine. Your OS9 machine will continue as your link to the future; an understandable and lovable link, at that. Best to all - Kevin" (end quote)

Another posting to the NET that might be of interest, dated Dec 7th 1989 and posted by Mike Knudsen :- "KMA... The motherboard will have plug-in sockets for four cards, each optional (more or less) at extra cost:

(1) 6809 with GIME and 512k RAM to run CoCo3 OS9L2 programs. Actually the GIME is an empty socket -- you steal your CoCo3's GIME and plug it in! This gets around Tandy legal problems (boy will they get a lot of GIME "repair part" orders).

Likewise the RAM area is empty, and you plug in your stock Tandy or 3rd-party 512k daughter board on that. And I assume your DISTO 1-meg if ya got it. Probably this board also has some stock CoCo peripheral hardware -- RS232, mouse/stick, and a socket to stick your disk controller or MPI into.

(2) SCSI interface board for floppy and hard drives, with DMA. This is for the 68k side, but might also be useable by the 6809 for users who don't already have disk controllers or who want the extra no-halt DMA performance.

(3) 68070 board with the Phillips VSC chip. If you get this you must take OSK, probably in ROM. Maybe you have to take this board and OSK to buy a CoCoPro at all. The political motivation here is to get us all to move to OSK, so you get it whether you want it or not. Anybody here really not want it? Aw c'mon...

Not clear whether you can plug your old 512k boards into this one. All video outputs and modes will be included. Hopefully you can switch between OS9 and OSK windows with the good old CLEAR key. Early on, this board will be able to take L2 window grafix ESCape commands from the 6809. I expect this will wean Ultimuse away from the VDG screens at last, and MVCanvas will be a pleasure to operate.

The big board will mount in a PC case and use a PC keyboard and power supply. It seems that the different daughter boards can share some of the memory space. That's it for now -- Mike K." (end quote)

So there is something to think about for this month. If and when, we see a CoCo4, KMA or whatever, here in Australia, only time will tell.

Regards, Gordon Bentzen

AUSTRALIAN OS9 NEWSLETTER

TAX, INCOME and INVESTMENT.
A spreadsheet template by David Eaton.

Editor's Note:- This Dynacalc Spreadsheet template and the accompanying article have come to us from David Eaton from Canberra. It is articles like this that will keep this newsletter going. Thank you very much, David.

Aim of the Programme

This Dynacalc programme is designed to:-

- (1) Summarise all necessary information for preparation of Income Tax returns for my wife and myself.
- (2) Summarise Total Income including reinvestments and compare actual Cash Receipts with Budgeted Cash Receipts.
- (3) Show Investment movements during the year and summarise the nett change in Investment.

Listings

There are two listings:-

The first listing is the plan or design on which the programme is based, the second listing shows the headings and the first two rows of an actual tabulation for a specific year. The first row shows the formula used in place of actual values where this is possible (i.e. where the values are calculated by formulae and are not directly entered values.)

Detailed Comments column by column

Column A

The first column is headed H'k or Housekeeping, which is a term I use for a column containing Indices to be used in 'QIF()' formulae in the body of the programme. In this case '0' signifies income in my name, and '1' income in my wife's name.

Columns BCD

Three columns used to contain a description of the Income Source e.g. name of investment.

Column E

Number of units in a Trust, number of shares in a Company or nominal value of a fixed Investment.

Column FGH

Provides four columns for entry of up to four payments a year from an income source. More or less columns can be provided as necessary. Enter all receipts whether income received in cash or reinvested. Exceptions are Medicare payments and income tax payments. Cash or credit from sale or maturity of investments should be included.

Column J

Sum of the previous four columns using 'SSUM(Fx...Ix)' where 'x' stands for row number. This formula can be replicated down the column as required.

AUSTRALIAN OS9 NEWSLETTER

Column K

Enter values for investments sold or reaching maturity.

Column L

Current income. $Jx-Kx$

Column M

Dividends or other income reinvested.

Column N

This is the actual cash income received. The formula $Jx-Kx-Mx$ represents total income less sales and reinvestment.

Column O

Budget Receipts: calculated from the previous year modified for any known changes.

Column P

For salary or superannuation tax is deducted at source. Enter here.

Column Q

Tax free income: some income, e.g. from Property trusts, has a tax free component. Enter here.

Column R

Input Credits. These are usually shown clearly on the payment slip from the Company or Trust but can be calculated.

Columns ST

The input credits are allocated to individuals from the formula @IF()

Column U

Total taxable income calculated with the formula $Jx-Kx+Px-Qx+Rx$.

Columns VW

Again allocation of tax between individuals.

Column X

Any new investment made during the year.

Column Y

Nett investment derived from $-Kx+Mx+Xx$

Column Z

Franked investment: Entered from the payment slip from Company or Trust.

AUSTRALIAN OS9 NEWSLETTER

Column AA

Unfranked investment: Required for the income tax return and derived from the formula $\text{@IF}(Zx>0, Jx-Kx-Zx, 0)$. This is not used in calculating taxable income.

INCOME, TAX AND INVESTMENT

Col	Column Heading	Formula or Value	Description
A	House Keeping	0 or 1	0 for TDE, 1 for DGE
BCD	Item	Name	Income Source
E	Nos or Nom value	Number	Number of Units or Nominal Value
F	Income Received	Values	All monies received or credited
J	Total Income	$\text{@SUM}(Fx...Ix)$	Total receipts and credits
K	Sold	Values	Investments sold or reaching maturity
L	Nett Income	$Jx-Kx$	Current Income
M	Reinvestment	Values	Dividends or other income reinvested
N	Nett Receipts	$Jx-Kx-Mx$	Actual cash income
O	Budget Receipts	Values	Previous Year's Income with known changes
P	Tax Deducted	Values	Tax deducted at source
Q	Tax Free Income	Values	Shown on payment slip
R	Input Credits	$(Zx/.61)*.39$	Preferably use figure given by Company
S	Input CR DGE	$\text{@IF}(Ax=1,Rx,0)$	Individual input credits
T	Input CR TDE	$\text{@IF}(Ax=0,Rx,0)$	Individual input credits
U	Total Taxable	$Jx-Kx+Px-Qx+Rx$	Total Taxable Income
V	Ttl Taxable DGE	$\text{@IF}(Ax=1,Ux,0)$	Individual Tax Total
W	Ttl Taxable TDE	$\text{@IF}(Ax=0,Ux,1)$	Individual Tax Total
X	New Investment	Values	Investment during year
Y	Nett Investment	$-Kx+Mx+Xx$	Nett increase or decrease
Z	Franked Invest	Values	From dividend slip
AA	Unfranked Inv	$\text{@IF}(Zx>0, Jx-Kx-Zx, 0)$	Required for Income Tax Return

AUSTRALIAN OS9 NEWSLETTER

General Comments

Four columns are provided for Income Received (Columns FGHI), but only one column for Tax Deducted or Input Credits though a number of entries may be necessary during the year. Dynacalc provides a neat way of adding new entries to existing ones by typing '#+new entry' (where 'new entry' is the new value to add (Ed)). Extra columns can be added if desired, but I have tried not to make the tabulation too unwieldy.

Sub headings can be provided in the 'Item' column corresponding to those used on the Income Tax return, e.g. Fixed Interest, Equity, Trusts.

This programme has been developed to meet my own specific need: other people will have different needs, but maybe this programme will help to give them a flying start in developing their own answers.

David Eaton.

MVCanvas a software review by Bob Devries

Package: MVCanvas
Source : Game Point Software
PO Box 6907
Burbank, CA, 91510-6907
USA
also : Nick Marentes
PO Box 551
Garden City
Qld, 4122
A\$35.00 (Jan '90)

Finally someone has written a drawing programme for OS9 along the lines of CocoMax III or ColorMax. The results of this first real attempt at drawing with pull-down menus and icons and such are very pleasing indeed. The programme works under the Multi-Vue system and uses the 'WindInt' pull-down menus and screens.

The pull-down menus at the top of the screen are:-

FILES EDIT GOODIES TOOLS FONT

Of course the Multi-Vue 'close gadget' is there at the extreme left of the menu bar. The contents of each pull-down menu look like this:-

FILES	EDIT	GOODIES	TOOLS	FONT
New Canvas	Undo	Cycle Color	Font Default	
Load	Cut to Disk	Cycle Speed	Plain	
Load IMG	Copy	Swap Color	Bold Off	
Save	Paste	Edit Brush	Transparent Off	
Save As...	Clear	640x200x4	Reverse Off	
Print	Show	Grey Scale	Underline Off	
Squash On	Invert	Load Font	Proportion Off	
Set Pic Dir	Flip Horizontal			
Set Execute	Flip Vertical			
Set Printer	Load Clip			
Directory				
About...				

AUSTRALIAN OS9 NEWSLETTER

The only menu I have not shown is the TOOLS menu, because the author, Mike Haaland, has made this a graphics buffer with the colours, patterns, and tool icons in it, and it is a bit hard to draw with a text editor. The programme performs very well indeed, and does everything that the menus suggest. Some items may be a little slower than its RSDOS counter-part, but then, RSDOS does not do any multi-tasking either. As I am writing this, I have a copy of MVCanvas running in 'W1' which is set up as a graphics window, as well as which I have a 32x16 VDG screen.

MVCanvas saves and loads 'VEF' format files (I haven't found out what that stands for yet), and conversion programmes are available to convert from CM3, MGE, DS69 formats to VEF. There are also some public domain programmes available to convert GIF files to VEF. MVCanvas will also read files written by Nick Marentes' RASCAN Digitiser (IMG) and so allow those people who own those to modify, re-colour, and title pictures from real-life.

MVCanvas uses standard OS9 Fonts file, so the files in your 'SYS' directory called 'stdfonts' could be used. There are also a few fonts converted from the RSDOS Graphicom II programme which I converted over to OS9 using Kevin Darling's programme from his book 'Inside OS9 Level II'. I think other fonts files could probably be converted also, especially those from, say, CocoMax III.

To run the programme from a hard disk, it is simply a matter of creating the necessary directories for the files, and copy all the files to them. MVCanvas has not been hard-coded for ANY directory names, only the file 'toolbox' must be either in the current working directory, or in '/DD/SYS' which is not an unreasonable request. If the toolbox file is not found, MVCanvas exits cleanly after displaying a message to tell you of the problem.

The only real problem I had with the programme, was with the printer drivers, none of which worked properly with my BMC BX-1000 printer. I feel, however, that this would not affect most people, as this printer is a bit of an odd one any-way. Further to this though, Mike Haaland will write a printer driver for you if your printer is not supported, if you send him the details of your printer.

The TOOLS menu contains these tools icons:-

SCROLL This allows the editing of the full 320 x 200 or 640 x 200 picture.

TEXT This allows you to print text on the screen in a previously selected font.

GRAB This allows you to select up to one quarter of the screen for editing purposes, and places it into the ClipBoard file.

FATBITS This allows a magnified view of a selected portion of the screen.

RADIANS This produces the familiar star shaped patterns.

LINE produces straight lines of course.

PENCIL is used for free-hand drawing.

BAR produces a filled box.

BOX produces an empty box.

ELLIPSE produces an ellipse of any size.

CIRCLE produces a circle.

SPRAYCAN sprays dots over the screen.

FILL fills the selected area with the paint colour.

BRUSH gives free-hand drawing with a larger tool.

AUSTRALIAN OS9 NEWSLETTER

STAMPS allows you to make a rubber stamp of an area of the screen so you can copy it to another part of your picture.

All these tools are represented by pictures (icons) and can be selected by clicking the mouse button while pointing to them. As well, there are sixteen colours and eight patterns to select from. There is also a facility to change a palette colour to any one of sixty-four colours.

I found MVCanvas easy to use, simple to install, and above all, good value for money. I can heartily recommend it.

Bob Devries

Module identification. by Bob Devries.

Don Berrie and I decided that it would be a good idea to print the module idents of the system modules that we are currently using, so that others can see whether they are up to date with their OS9 versions. These listings were produced by IDENT (and then modified to shorten them a little). They are all level two modules. You'll notice that some have comments beside them regarding modifications and other versions. You'll notice that I have not included any device descriptors, as these will change (and should) with each individual system.

Header for: OS9p2
Module size: \$0CAE #3246
Module CRC: \$47B370 (Good)
Edition: \$11 #17

Header for: IOMan
Module size: \$09F3 #2547
Module CRC: \$FD1FEA (Good)
Edition: \$0C #12

Header for: Init modified for number of open paths.
Module size: \$002E #46
Module CRC: \$6B0456 (Good)
Edition: \$43 #67

Header for: CC3Go
Module size: \$01AE #430
Module CRC: \$1006FE (Good)
Edition: \$05 #5

Header for: Clock Special version for CRC SCII with hardware clock.
Module size: \$027F #639
Module CRC: \$ED15E5 (Good)
Edition: \$03 #3

Header for: RBF
Module size: \$122E #4654
Module CRC: \$EFBE13 (Good)
Edition: \$10 #28

Header for: OCHDisk Special version for CRC SCII SASI hard disk interface.
Module size: \$02F0 #752
Module CRC: \$A1199D (Good)
Edition: \$04 #4

Header for: CC3Disk Special version for CRC SCII No-halt controller.
Module size: \$04EA #1258
Module CRC: \$A27B77 (Good)

AUSTRALIAN OS9 NEWSLETTER

Edition: \$A3 #163

Header for: Rammer Public domain Ramdisk driver.
Module size: \$0129 #296
Module CRC: \$15C571 (Good)
Edition: \$03 #3

Header for: SCF Modified with Kevin Darling's command line edit functions.
Module size: \$06A8 #1707
Module CRC: \$7E683A (Good)
Edition: \$0D #13

Header for: CC3IO Patched for some bugs.
Module size: \$0C64 #3172
Module CRC: \$14BA44 (Good)
Edition: \$10 #16

Header for: WindInt Patched.
Module size: \$1D6C #7532
Module CRC: \$744DB8 (Good)
Edition: \$12 #18

Header for: ACIAPAK Modified to work on CRC SCII 4 in 1 board RS-232 board.
Module size: \$0385 #949
Module CRC: \$C1EA10 (Good)
Edition: \$0C #12

Header for: PRINTER
Module size: \$017A #378
Module CRC: \$CC3EA4 (Good)
Edition: \$0C #12

Header for: Parallel Special parallel printer driver for CRC 4 in 1 board.
Module size: \$0095 #149
Module CRC: \$A70A4C (Good)
Edition: \$06 #6

Header for: VDGInt
Module size: \$0CE5 #3301
Module CRC: \$7693A7 (Good)
Edition: \$01 #1

Header for: PipeMan
Module size: \$0219 #537
Module CRC: \$AD6718 (Good)
Edition: \$04 #4

Header for: Piper
Module size: \$0028 #40
Module CRC: \$5B2B56 (Good)
Edition: \$02 #2

Header for: Pipe
Module size: \$0026 #38
Module CRC: \$CC06AF (Good)
Edition: \$50 #80

AUSTRALIAN OS9 NEWSLETTER

Header for: Printer 1 Modified from Level One.
Module size: \$0160 #352
Module CRC: \$0B030D (Good)
Edition: \$06 #6

Header for: Init Original LII Version 02.00.01
Module size: \$002E #46
Module CRC: \$0E2322 (Good)
Edition: \$43 #67

Header for: CC3Disk Ditto
Module size: \$0454 #1108
Module CRC: \$759161 (Good)
Edition: \$09 #9

Header for: SCF Ditto
Module size: \$05E3 #1507
Module CRC: \$F946CA (Good)
Edition: \$00 #13

Header for: CC310 Ditto
Module size: \$0036 #3126
Module CRC: \$F737C2 (Good)
Edition: \$10 #16

Header for: VDGInt Ditto
Module size: \$0CE5 #3301
Module CRC: \$11ED90 (Good)
Edition: \$01 #1

Header for: GrfInt Ditto
Module size: \$0D3D #3389
Module CRC: \$6E4441 (Good)
Edition: \$0E #14

Header for: ACIAPAK Ditto
Module size: \$03B5 #949
Module CRC: \$C1EA10 (Good)
Edition: \$0C #12

Header for: S10 Ditto
Module size: \$0185 #389
Module CRC: \$915957 (Good)
Edition: \$09 #9

Header for: Clock Ditto
Module size: \$01EE #494
Module CRC: \$D28AFD (Good)
Edition: \$09 #9

AUSTRALIAN OS9 NEWSLETTER

VARIATIONS ON A DISK ZAP SCENE INCLUDING MODIFICATIONS TO COCO OS9 DISK ZAP

Some other minor bugs in my disk zapper programme ZAP, have come to light. They are not really faults with my programming, but rather, result from problems with the way that some of the Basic09 number manipulation functions handle large non-integer numbers.

The version of the disk zapper which is currently circulating throughout the user group is starting to get "dated", and these patches will be the last to be made to it, as I can no longer keep supporting that archaic version of the code.

One of my recent projects, with a great deal of help from Bob Devries, has been a major upgrade to this disk zapper programme (the newest Version is 1.19), but still written in Basic09. Amongst the changes made are some advanced techniques for opening windows, all of the bugs fixed (as far as we know), so that the programme can be used on any sized hard disk, a major improvement in the speed at which the programme works (by about 50 %), the ability to use (automatically) a parallel printer interface, if present and lots of other minor changes. The size of the code of the packed version has increased from Hex. 2047 to Hex. 3937.

Because of the not inconsiderable time we have invested in the rewrite project, we have decided that we will only release the packed version of the programme into the public domain. And therefore, we will only be able to distribute it via the National User Group Library. The usual postage and copying fees apply, but you will be able to distribute it freely to other users. We are currently working on version 2.00. All future revisions (including 2.00) will be written in C for even faster operation.

Improvements and enhancements to the C versions will possibly include an improved user interface (mouse driven ?), logical sector 0 decoding into plain text, a much improved calculator function, the ability to work with files and tracks as well as sectors, a means of editing (with reference to files) the sector allocation tables, ascii editing of disk sectors, automatic CRC updates, and all sorts of other goodies. We will even be trying to include such things as file undelete functions, the ability to read and repair damaged disks, and perhaps even single track formatting!! So keep reading the newsletter, because you will find out about it first in these pages.

Now for the patches to the original version. Even with these patches, if you have a hard disk, you will still not be able to access sectors greater than \$FFFF. Sorry about that, but the changes necessary to this version to allow it to be able to do that, are just too involved to justify the space in the newsletter.

Firstly, there is a problem caused by the way that the VAL function decodes large numbers.

When a string representing a number which is larger than 32768 (32K) is passed to VAL, it interprets it as a two's complement number, and therefore gets a negative result! Hardly what one needs to reference a sector number on a disk.

Normally, this would never matter, because the maximum number of sectors on an 80 track double sided disk is only 2880. On a hard disk however, it is a different matter. My hard drive has 78720 sectors (20 meg) and therefore the problem can arise.

The following changes (marked in boldface) to the Getsec Procedure will avoid the problem, and allow you to access sectors from \$8000 to \$FFFF. The programme will still "hang" if you try to access sectors \$10000 and above. (Incidentally, the new, packed version doesn't hiccup at all when reading sectors up to \$FFFFFF, and that's as far as you can go with the current versions of OS9 anyway).

```
PROCEDURE getsec
BASE 0
PARAM wpath:BYTE
PARAM maxblock:REAL
PARAM blkno:REAL
PARAM secdat(256):BYTE
DIM hblkno:STRING(25)
DIM max0,max1:INTEGER
```

AUSTRALIAN OS9 NEWSLETTER

```

DIM i:INTEGER (* New line inserted *)
max0=0
IF maxblock>65535. THEN
max0=INT(maxblock/65536.) (* Note 65535 changed to 65536 *)
max1=MOD(maxblock,65536.) (*      *      *      *      *)
ELSE
max1=maxblock
ENDIF
1 PRINT #wpath,"SECTOR NUMBER (max:$";
PRINT #wpath USING "H2",max0;
PRINT #wpath USING "H4",max1;
PRINT #wpath,") :";
INPUT #wpath," ",hblkno
(* hblkno="$"+hblkno *** This line now deleted *)
ON ERROR GOTO 1
(* blkno=VAL(hblkno) *** This line now deleted *)
blkno=0 (* Four      *)
FOR i=LEN(hblkno) TO 1 STEP -1 (* new      *)
blkno=blkno+VAL("$"+MID$(hblkno,i,1))*16^(LEN(hblkno)-i) (* lines      *)
NEXT i (* inserted      *)
IF blkno>maxblock OR blkno<0 THEN
GOTO 1
ENDIF
RUN winclose(wpath)
END

```

All that this modification does is to parse the input hexadecimal string into individual single digit hex numbers, then uses VAL to decode these to decimal, and then performs the necessary multiplication to derive a real number as the final result.

The second problem stems from the fact that the BasicOS9 PRINT USING "H#" (where # is a number) function, does not reliably decode non-integer numbers, and, as well, cannot handle numbers greater than 65536 (64K). The following patches to the Scn Procedure overcomes the problems.

```

PROCEDURE scn
ON ERROR GOTO 1000
BASE 0
PARAM wpath:BYTE
PARAM NAME:STRING[4]
(* PARAM iblkno:INTEGER *) (* Line deleted *)
PARAM blkno:REAL (* New line inserted *)
PARAM secdat(256):BYTE
DIM COUNT:REAL
DIM i:INTEGER
DIM PAGE:BYTE
DIM outstr:STRING[16]
DIM tblkno(2):INTEGER (* New line inserted *)
PRINT #wpath,CHR$(#0C)
PAGE=0 \outstr=""
COUNT=0
PRINT #wpath,"DEVICE : "; LEFT$(NAME,LEN(NAME)-1); " "; "SECTOR : $";
(* PRINT #wpath USING "H4",iblkno *) (* Line deleted *)
tblkno(0)=INT(blkno/65536.) (* Four      *)
PRINT #wpath USING "H2",tblkno(0); (* New      *)
tblkno(1)=INT(MOD(blkno,65536.)) (* Lines      *)
PRINT #wpath USING "H4",tblkno(1); (* Inserted *)
PRINT #wpath,
PRINT #wpath,"Rel  0 1 2 3 4 5 6 7 8 9 A B C D E F "

```

AUSTRALIAN OS9 NEWSLETTER

```
PRINT #wpath,"Addr -----"
FOR i=0 TO 255
IF COUNT=0 THEN
PRINT #wpath USING "h2>",PAGE;
PAGE=PAGE+1
PRINT #wpath,"  ";
ENDIF
PRINT #wpath USING "h2>",secdat(i);
COUNT=COUNT+1
IF secdat(i)<$21 OR secdat(i)>$7A THEN
outstr=outstr+."
ELSE
outstr=outstr+CHR$(secdat(i))
ENDIF
IF COUNT=16 THEN
PRINT #wpath
COUNT=0
ENDIF
NEXT i
PRINT #wpath
END
1000 RUN closerr(wpath)
END
```

You will also need to modify one line in the main part of the programme in order to pass the correct sector number to the screen print procedure.

```
PROCEDURE zap
```

```
11 RUN scn(wpath,NAME,blkno,secdat) (* Parameter name changed iblkno=blkno *)
```

A similar routine within the main procedure is used to print a sector to the printer. Similar changes to that part of the main Procedure are required in order for the print routine to function correctly. I leave it to you to make any changes that are required in that part of that Procedure.

oooooooooooo0000000000oooooooooooo

YET MORE PATCHES

While on the subject of patches, the printer gremlins got to the shellscript programme that was published in the last version of the newsletter. (I went through and altered one line by hand on every copy !!!). Unfortunately, however I missed noticing that another line was also missing. You will need to add the line:

```
chd %1
```

immediately prior to line 16. (line 16 reads : *Loop1)

Otherwise, as you may have found out you will have problems finding your files.

If you have any problems with this shellscript, or the patches mentioned in the article above, please do not hesitate to call me.

Cheers, Don Berrie. (07) 375-3236.